AD-A076 965

TECHNICAL LIBRARY

AD

AD-E400 359

TECHNICAL REPORT ARSCD-TR-79006

COMBAT SIMULATION USING BREACH COMPUTER LANGUAGE

RICHARD KWATNOSKI

NICHOLAS SCORS

SEPTEMBER 1979

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
FIRE CONTROL AND SMALL CALIBER
WEAPON SYSTEMS LABORATORY
DOVER, NEW JERSEY

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

Destroy this report when no longer needed. Do not return it to the originator.

The citation in this report of the names of commercial firms or commercially available products or services does not constitute official endorsement or approval of such commercial firms, products, or services by the United States Government.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	ARSCD-TR-79006
2. GOVT ACCESSION NO.	
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle)	COMBAT SIMULATION USING BREACH COMPUTER LANGUAGE
5. TYPE OF REPORT & PERIOD COVERED	
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s)	Richard Kwatnoski Nicholas Scors
8. CONTRACT OR GRANT NUMBER(s)	
9. PERFORMING ORGANIZATION NAME AND ADDRESS	ARRADCOM, FC&SCWSL Systems Div (DRDAR-SCS-M) Dover, NJ 07801
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	1L162617AH19TA03
11. CONTROLLING OFFICE NAME AND ADDRESS	ARRADCOM, TSD ATTN: STINFO Div (DRDAR-TSS) Dover, NJ 07801
12. REPORT DATE	SEPTEMBER 1979
13. NUMBER OF PAGES	38
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	ARRADCOM, FC&SCWSL Systems Div (DRDAR-SCS-M) Dover, NJ 07801
15. SECURITY CLASS. (of this report)	UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Simulation
Combat simulation
Systems analysis
Stochastic modeling
Computer language
BREACH
Urban warfare
MOBA
MOUT

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report highlights and summarizes certain light weapon system analysis efforts conducted during FY78 and 79. It also evaluates the effectiveness of existing, developmental, or conceptual small arms weapons in a combat environment. This study presents a method for analyzing light weapon systems through the use of realistic combat simulation models.

20. (Cont'd)

Computer models were constructed using the Battlefield Related Evaluation and Analysis of Concepts and Hardware (BREACH) simulation language. BREACH is a high resolution, combat simulation and weapon system analysis computer language. Two types of models were constructed: a stochastic duel and a dynamic engagement model. The duel model validates the BREACH approach by comparing results with mathematical solutions. The dynamic model shows the capability of the BREACH language to perform complex weapon system analysis.

The major conclusion drawn from the analysis is that the BREACH language offers a unique capability for conducting combat simulations for a broad range of weapon analysis problems.

ACKNOWLEDGMENTS

The approach to building the BREACH simulation capability within ARRADCOM was one of exceptional interdepartmental cooperation. In addition to the authors, the team which conducted the BREACH analysis included LT Rocco Antonelli, Systems Division, and Mr. James Steiner, Armament Division, FC&SCWSL; and Mr. John Tobak, Scientific and Engineering Application Division, Management Information Systems Directorate.

The authors are particularly grateful to Mr. Steiner for his weapons system analysis, the high technical caliber of which deserves special acknowledgment. Mr. Steiner, the most knowledgeable BREACH analyst in FC&SCL, was responsible for most of the coding.

Special acknowledgment is also due Mr. Tobak for his exceptional software support of the BREACH language, including converting Version 8 from Univac to CDC, affecting the necessary software changes, and initiating interactive graphics.

Mr. Louis Barbarek, of the ITT Research Institute, conducted the "BREACH Orientation Workshop" for ARRADCOM personnel and instructed the Light Weapons Systems Analysis team in the use of the language.

TABLE OF CONTENTS

TABLES

FIGURES

INTRODUCTION

The objective of this report is to illustrate a method for conducting combat simulation studies of light weapons systems using Battlefield Related Evaluation and Analysis of Concepts and Hardware (BREACH) computer language. The program was initially funded during FY78 from the Light Weapon Tech Area of Project AH19 under Trade-off Analysis/Urban Warfare Weapons. The original program objective was to evaluate the effectiveness of existing, developmental, or conceptual small arms weapons in an urban environment. During FY79 funding was received from the same technical area, but for more generalized systems analysis support of light weapons projects. This explains why the report emphasizes urban warfare simulations. During late FY78, a small amount of additional funding was received from the Battlefield Systems Integration Directorate, DARCOM, and a contract was let with the IIT Research Institue for conducting the "BREACH Orientation Workshop" at ARRADCOM.

Simulation studies are appropriate when a system is stochastic (i.e., part of the response is random in nature) or when straight forward mathematical methods fail due to equation complexity introduced by the various interactions of the parameters. In establishing a computer simulation model there is a tendency to attempt to incorporate too much detail in the belief that the more detail, the more realistic the model. However, the more detail there is the more problem areas there are. These include:

Time and effort must be devoted to the observation of the preliminary characteristics of the system.

The programming and debugging effort must be increased.

The program running time and cost must be expanded.

In the attempt to construct combat models for light weapons analysis, the concept of a single, all encompassing model was rejected. Past efforts at constructing large, complex small arms models have led to either failure or dissatisfaction with the results. Rather, the concept of writing simple and responsive simulation models that incorporate the parameters of interest was selected. BREACH is ammenable to this concept since it is a language which facilitates model building and since its complexity is determined by the model builder. A search was conducted of the existing programming languages and programs that are appropriate to the task of small arms weapon system evaluation. BREACH was eventually selected because of its history of applications and acceptance and usage throughout the Defense community.

BREACH

Background

BREACH is a high resolution combat simulation and weapons systems analysis computer language. A general purpose language, it has been used by all the Services. Diverse applications have included mine, armored assault, electronic warfare, and urban warfare studies.

Actually, the language in its present form has evolved over nine versions during a period of almost 10 years. Original versions were produced for what is now MERADCOM for vehicle and minefield studies. Version 5 was produced for the Navy and Marine Corps for analyzing underwater mines. Version 6 added enhancements for studies of GATOR and ADAM mine systems. Version 7 was the first well-documented version with a Users, Analysts, and Programmers Manual published by the Joint Technical Coordinating Group for Munitions Effectiveness (JTCG/ME) (ref 1-3). It should be noted that up through Version 7, BREACH was directed toward mine studies. Unfortunately, the command syntax of BREACH reflected this in the sense that BREACH was a general purpose language with misleading mine-oriented syntax.

This problem became much less apparent with subsequent Versions of BREACH. Version 8 was produced for the US Air Force (Eglin AFB) and offered some additional, minor enhancements. These included: continuation cards, text output, improved detection routines, and added flexibility in vehicle maneuvering (ref 4, 5). Version 9 was the last major version of BREACH; and was produced for what was then the US Army Electonic Command, Fort Monmouth. At this point, BREACH became known as BREWS, which stands for Battlefield Related Electronic Warfare Simulation. Version 9 contained numerous, significant enhancements including: continuous terrain considerations, improved detection routines with line of sight capability, missile trajectory subroutines, and consideration of different kill levels (suppression, firepower, mobility and total) (ref 6-8).

BREACH, up through Version 7, was written for use on the Control Data Corporation (CDC) 6500/6600 series computers. Versions 8 and 9 were written for the Univac 1108 series of computers. Unfortunately, Versions 8 and 9 are not compatible with CDC computers. ARRADCOM primarily employs the CDC 6500/6600 computer series, except for a Univac machine at the CSL, Edgewood site. The Management Information Systems Directorate (MISD) had to therefore convert Version 8 to the CDC system. Verson 9, however, which is considerably more complex, would require a major effort for conversion. All BREACH modeling performed at the Dover site, ARRADCOM, was performed primarily with Version 7 and recently with Version 8.

The Language

The BREACH language is divided into three major phases: Executive, Control, and Input. The Input Phase is further subdivided into six minor phases: Environment, Object, Emplacement, Neutralization Device, Detection, and Vehicle. All coding in BREACH is done in a command format with optional parameter strings associated with the command. The language is based upon FORTRAN subroutines which are accessed and exercised via the BREACH commands.

The Executive Phase is the action part of the language which drives the simulation. Sample commands are listed below:

PATH: Builds table of events along a specified path.

MOVE: Moves vehicle along path.

DELIVER: Delivers neutralization devices (munitions).

LOCATE: Locates objects, vehicles and/or points on the map.

FIRE: Describes a direct or indirect firing table of times, distances, hit points, and probabilities of kill.

The Control Phase applies to all the other phases, and is of interest primarily to the programmer. It includes such general areas as file manipulation, I/O information control, central processor unit (CPU) time monitor, and random number generator seeding.

As mentioned previously, the Input Phase is subdivided into six minor phases, the first of which is the Environment Phase which may be thought of simply as "the map." It is primarily terrain description through which one may control detection, visibility, mobility, and elevation.

The Object Phase includes description of the characteristics of stationary objects and obstacles. Objects may be either active or passive. An example of an active object would be a mine; a passive object would be a building or a barrier.

The Emplacement Phase defines the emplacement of stationary objects either one-by-one or according to statistical distributions.

The Neutralization Device Phase describes munitions performance. Sample commands are as follows:

CIRCULAR: Describes effectiveness area of a circular shape.

LINEAR: Describes effectiveness area of a linear shape.

WEAPON: Describes a firing device with effectiveness area of discrete circular shapes.

DEMOLITION: Describes a manual neutralization method.

The Detection Phase describes both visual and analytic detection. Visual detection, as the name implies, characterizes the effectiveness of human (or animal) detection. Analytic detection describes and specifies the effectiveness of detectors having an analytically expressable probability of detection.

The final, minor phase of Input is the Vehicle Phase. This Phase is used to describe all moving objects--both mechanical and human--and their associated vulnerability/lethality.

STATIC DUEL

Background and Methodology

A two-sided static duel between an M16 rifle and Squad Automatic Weapon (SAW) in an urban environment was constructed using BREACH. The primary purpose of this effort was to initiate our urban warfare modeling. The secondary purpose was to construct a moderately complex BREACH computer model which could be compared to a known analytical solution.

Scenario

The scenario analyzed was as follows: an attacking soldier (Blue) employing the SAW weapon is located in the street. The SAW is bipod mounted and is fired in five round bursts. A defending soldier (Red) is located within a building 100 meters away. Red employs an M16 rifle which is fired in three round bursts from the prone position. Red initiates the engagement by firing two bursts within 10 seconds, with Blue then returning fire. The firing sequences then alternates between attacker and defender with a burst fired every 7½ seconds (fig 1).

Both Red and Blue are partially obscured; therefore, an upper torso target is presented to both firers. A hit probability for Blue and Red was calculated using the MAGUN Computer program (ref 9) based on weapon, firing mode, number of rounds per burst, range, and target presented area. Red's probability of hitting Blue (P(R)) at least once per burst is 0.10, and conversely P(B) is 0.24. Projectile time-of flighs were considered for dual kills. For simplicity, since both systems were assumed to employ the same cartridge (XM777), incapacitation probabilities were not considered.

A hit was assumed to be a kill and the engagement was terminated when a hit occured. A plot of Red's probability of being killed versus replication numer is shown in figure 2.

Validation

After constructing and exercising the simple duel, it was felt that validation of the BREACH model was necessary before proceeding with more complex analysis. Recognizing that this particular duel reduces to a Markov process where each event depends upon the outcome of the preceeding event, a validation procedure was suggested by Groves (ref 10) for fixed, or nonrandom, rate-of-fire duels. However, the procedure was modified to account for the nonrepetiveness of the first firing cycle (i.e. Red fires upon Blue twice before Blue can return fire). Using the following notation:

$P_N\ (R)$ = Probability of Red winning duel on the Nth shot.

where P(R) is used interchangeably with $P_1\ (R)$

$P_N\ (B)$ = Probability of Blue winning duel on the Nth shot.

where P(B) is used interchangeably with $P_1\ (R)$

$P_D\ (R)$ = Probability of Red winning duel.

$P_D\ (B)$ = Probability of Blue winning duel.

The following equations illustrate the probability of Blue winning the duel:

$$P_1\ (B) = (1-P(R))\ (1-P(R))\ P(B) = 0.1944$$

Recalling that Red fires twice before Blue returns fire. Blue's first shot is actually the third shot of the duel.

$$P_2\ (B) = (1-P(R))\ (1-P(B))\ (1-P(R))\ (1-P(R))\ P(B) = 0.1330$$

$$P_N\ (B) = (1-P(R))^{N-1}\ (1-P(B))^{N-1}\ (1-P(R))^2\ P(B)$$

$$P_D\ (B) = \sum_{N=1}^{\infty} P_N\ (B)$$

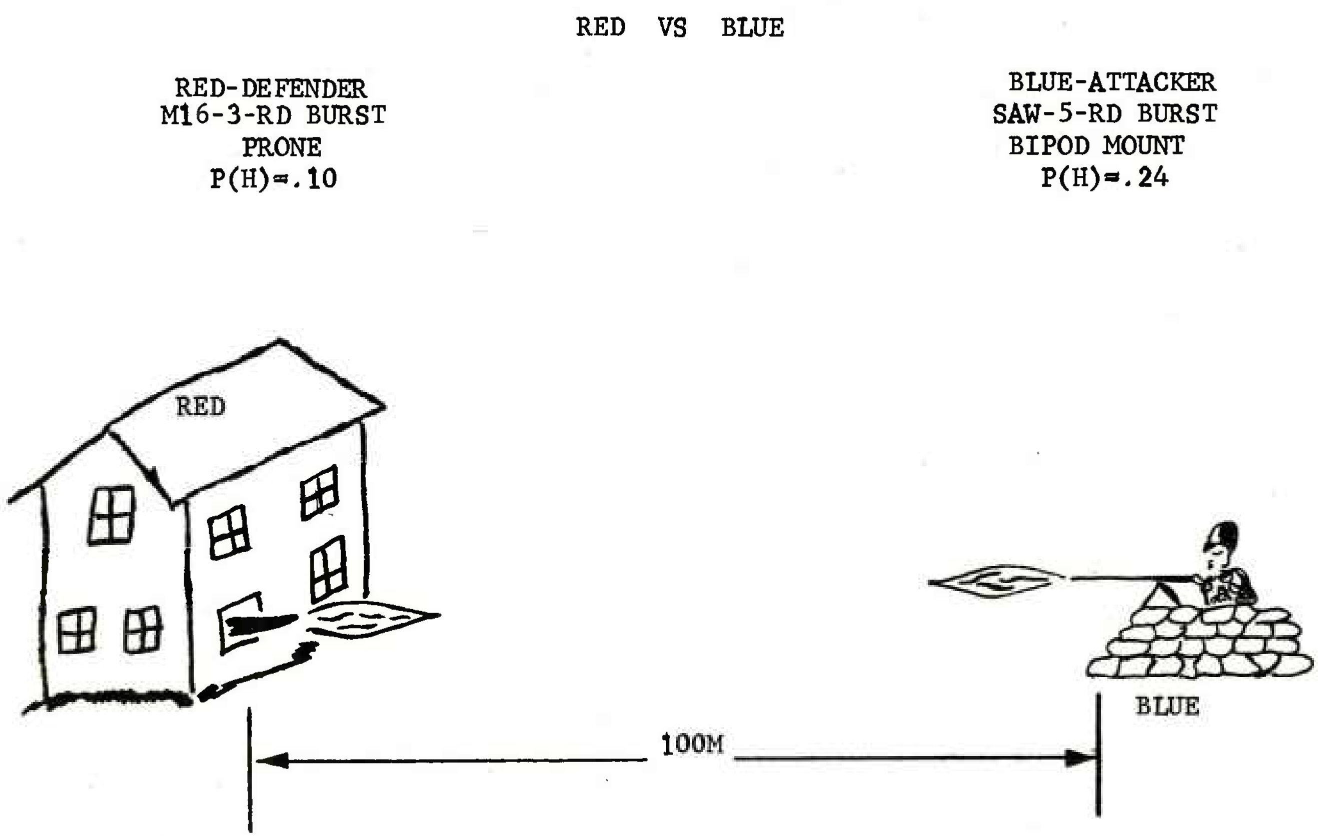

Figure 1. Two-sided duel-red vs blue.

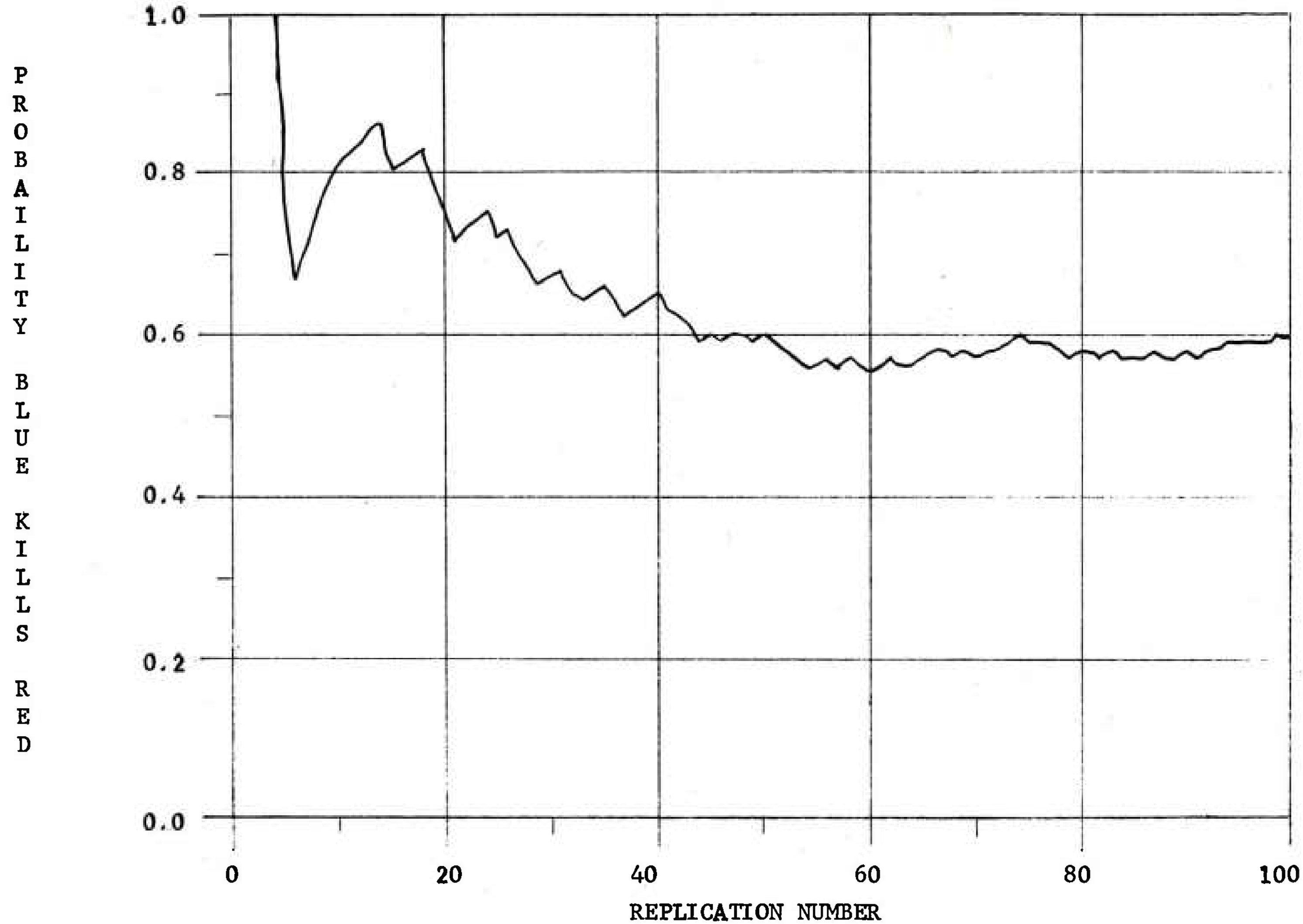

Figure 2. Probability of kill versus replication number.

$$P_D(B) = \frac{P(B)\ (1-P(R))^2}{1-(1-P(R))\ (1-P(B))}$$

$$P_D(B) = \frac{(0.24)\ (0.90)^2}{1-(0.90)\ (0.76)} = 0.615$$

Results

The results of this analysis indicate that the probability of the SAW (Blue) winning the duel is 0.615 (0.60 as approximated by the Simulation Study), even though Blue is fired upon twice by the M16 before returning fire. The SAW gains its advantage from the firing mode and mount employed. A comparison of the results of the BREACH simulation with the analytical solution shows that BREACH simulation models can be accurate and useful analytical tools.

This static duel model could be expanded to give information on:

- Weapon effectiveness
- Ammunition expenditure
- Engagement time

TANK DUEL SIMULATION

The tank engagements simulated were taken from a concept in an unpublished paper, "A Proposed Probabilistic Monte Carlo Analogue Concept," by Herbert N. Cohen, US Army Concepts Analysis Agency. The scenario of this engagement is as follows (fig 3): This is a two vs two engagement (N and N' vs M and M') where M and M' fire at N while N and N' fire at M.

A BREACH computer model was written to simulate the tank engagement with the following input variables: individual tank hit probabilities, maximum number of tank rounds, time between rounds, and delay time of first firing event. The probability of N being killed P(N) was used as the measure of record.

Four representative cases of the tank duel were investigated using both the BREACH computer model and analytical solutions. Using the BREACH model, each case was replicated one hundred times. A summary of the input variables and results for the four cases are summarized in table 1.

Representative analytical solutions to the two tank engagements investigated are as follows:

For case II the probability that N is killed is equal to the probability that N is killed by M' (0.10) plus the probability that N is killed by M (0.99)(0.09)). This probability is equal to 0.1891 (0.10+(0.99)(0.09).

For case III the probability that N is killed is equal to 1 minus the probability that N survives the four shots from M' ($(0.90)^4$) plus the probability that M survives ($(0.30)^8$) times the probability that M kills N(0.99). This probability is equal to 0.343965 $(1-(0.90)^4 + (0.3)^8 (0.99)]$. The close agreement of the BREACH model simulation with the analytically calculated P(N) validates the BREACH model and the accuracy of its output.

DYNAMIC ASSAULT MODEL

Background

This assault model is a dynamic simulation of a two-sided engagement. This model was constructed to illustrate the capabilities of a BREACH computer simulation model. While an urban warfare combat scenario was selected as an example, one can also apply this modeling approach to most types of high-resolution combat scenarios.

The dynamic assault model highlights the major featrues of the BREACH language. BREACH's structure and subroutines enable the programmer to model and modify his simulation with less coding and greater ease.

The scenario of the model is as follows: A defender (Red) located on the roof of a two-story building is assaulted by three attackers (Blue). The attackers use fire and movement for their assault. The defender first fires on the moving attacker and, after a suitable delay (10 sec) for target acquisition, the two stationary attackers engage the defender. This sequence is repeated until either the defender is killed or two (of three) blue attackers are killed (fig 5).

In this study the effect of varying the defenders weapon was investigated. In case 1 the defender uses the M16 rifle, firing a three-round burst from the prone position. In case 2 the defender fires a five-round burst using a SAW with a bipod mount. The two blue cover men fire three-round bursts from the M16 rifle in the prone position. Since both the Red defender and the moving Blue attacker are partially obscured, an upper torso target is assumed.

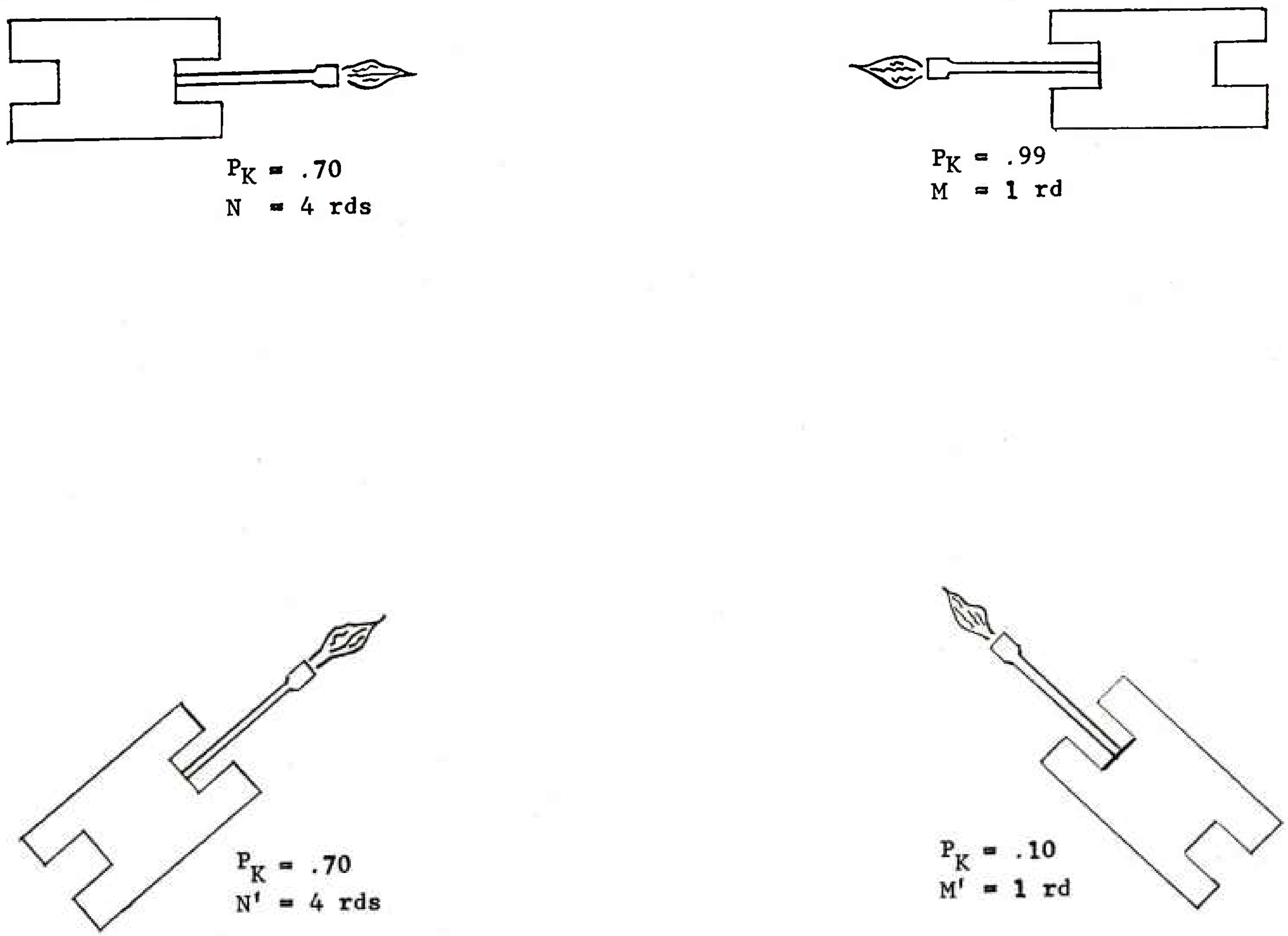

Figure 3. M vs N tank duel.

Table 1. BREACH tank duel simulation

		CASE		
	I	II	III	IV
PH (N)	0.70	0.70	0.70	0.875
N'	0.70	0.70	0.70	0.875
M	0.99	0.99	0.99	0.990
M'	0.10	0.10	0.10	0.100
RDS (N)	4	4	4	4
N'	4	4	4	4
M	1	1	1	1
M'	1	1	4	1
TOF(N,N'M,M')	0			
TER (N)	0.001	20.00	0.001	0.001
N'	0.001	20.00	0.001	0.001
M	10.000	10.00	10.000	10.000
M'	0	0	0	0
TTF (N)	0	0	0	0
N'	0	0	0	0
M	10	10	10	10
M'	0	0	0	0
P(N) Actual	0.10	0.19	0.34	0.52
P(N) From Simulation	0.12	0.18	0.31	0.54

where

PH = Probability of hit
RDA = Max no. of rounds
TOF = Time of flight
TER = Time between rounds
TTF = Delay time of first firing event
P(N) = Probability N is killed

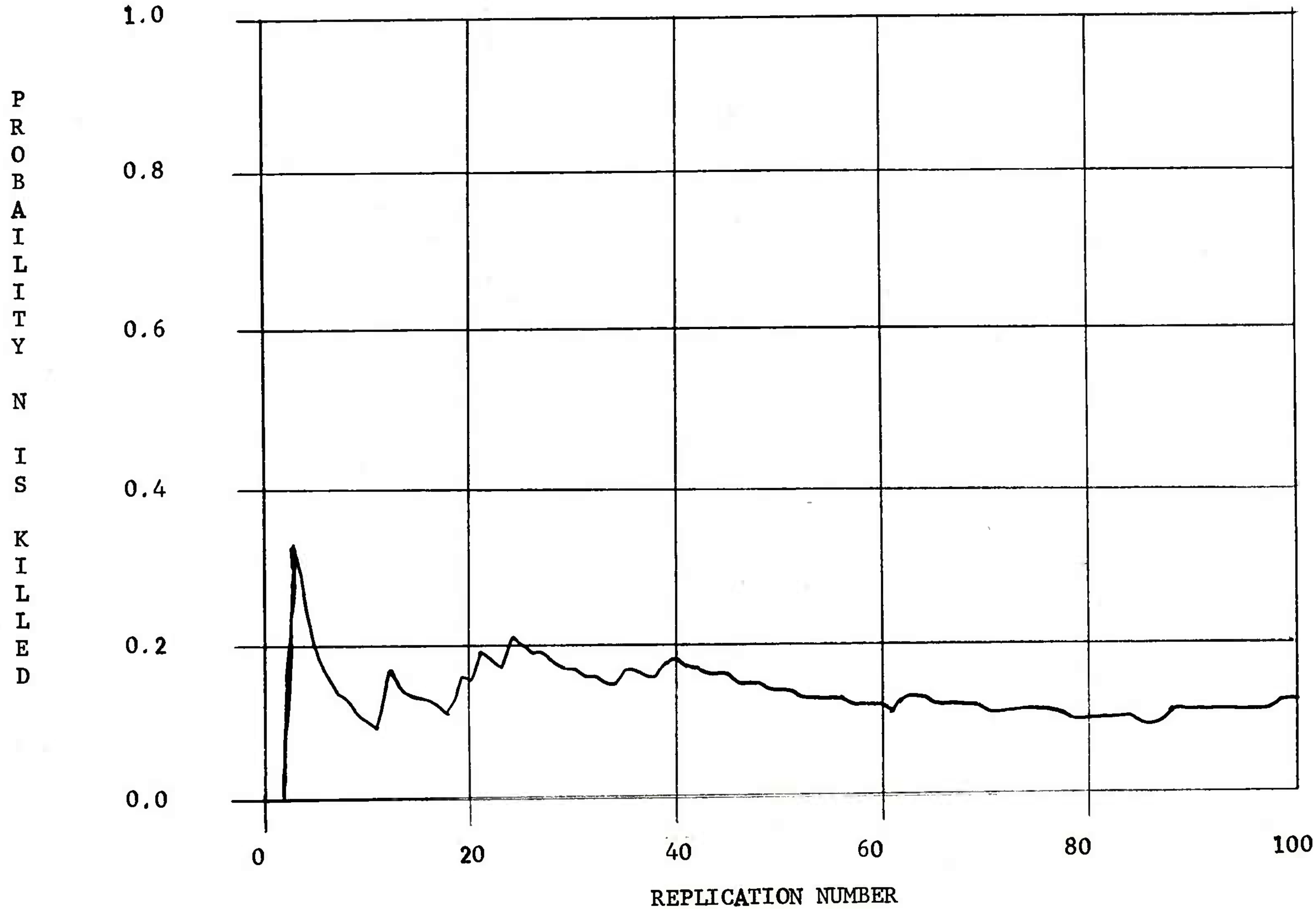

Figure 4. Probability of kill versus replication number.

Features

The assault model allows one to set the following variables:

- Individual firing times
- Weapon firing mode and hit probabilities
- Movement distances with subsequent exposure times
- Individual reaction times including target acquisition times.

The following output is obtainable from the assault model:

- Individual killed and location
- Time of kill
- Killing weapon and person

The following conditions/rules apply to this model:

- All hits are kills
- There are no suppression effects
- Slant angle is not taken into account.

The dynamic assault model illustrates the following features of BREACH simulations:

- Dynamic hit probability calculations. Using BREACH's Weapon and Effect Commands, a hit probability versus range curve is generated. This is an exponential decay hit probability curve.

- Map Construction. A basic map with a grid system was constructed. BREACH facilitates the creation and placement of such items as buildings, streets, obstacles, and soldiers. A soldier can be moved from obstacle to obstacle by use of the MOVE command. The BREACH program keeps a record of the individual soldier's status, location, and time.

- Model Clock. Clock time is maintained by the BREACH program allowing the analyst/programmer to compare ending times for firing events and to set status and counters based on event outcome clock times.

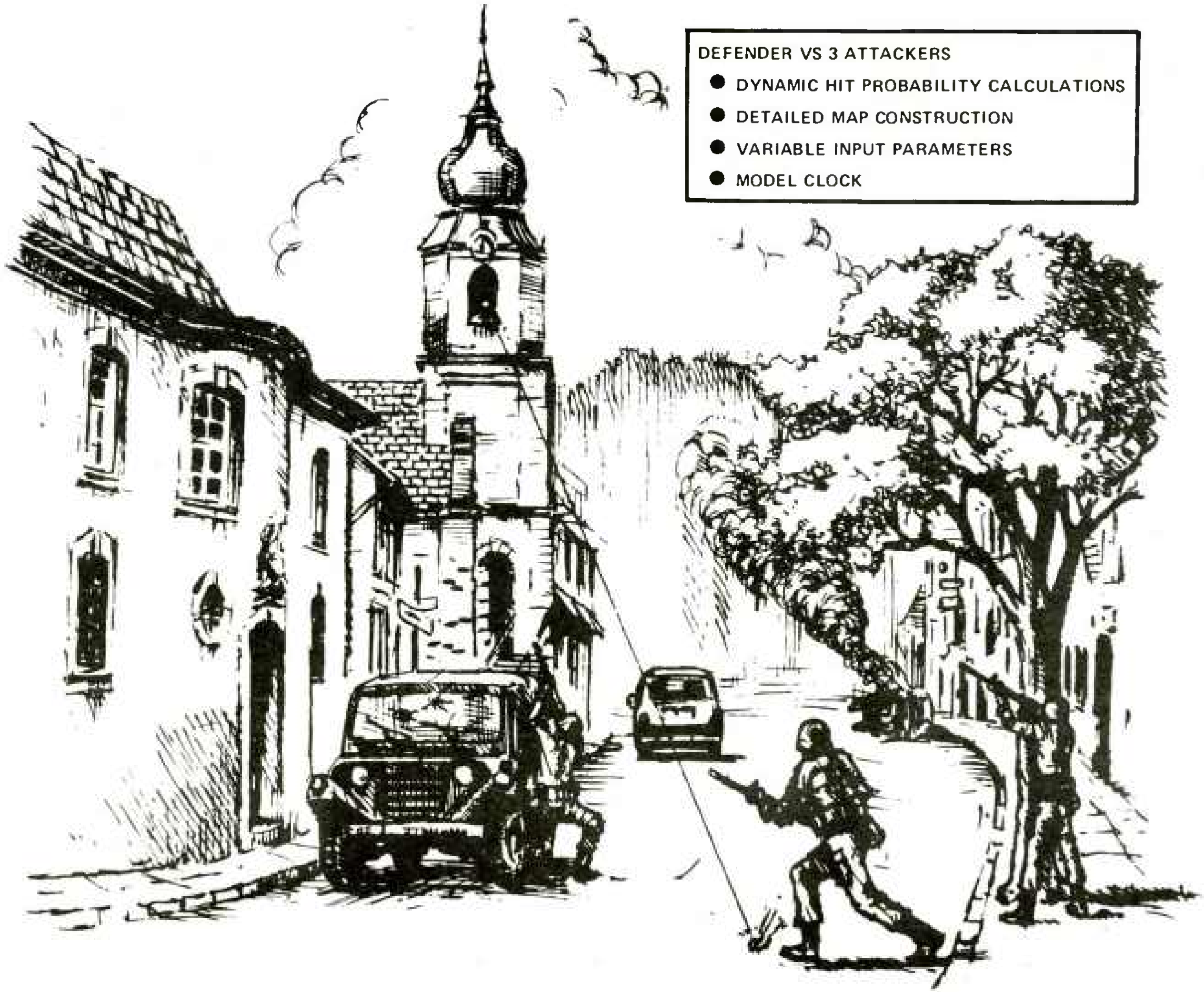

Figure 5. Dynamic assault model.

The dynamic assault model was used to investigate the effect of varying the defender's weapon. The model was exercised for 12 replications with both the M16 (case 1) and the SAW (case 2).

The following were input parameters to the model:

- Initial engagement range (500 m)

- Movement increments (25 m)

- Attackers velocity (5 m/sec)

- Firing event frequency (one event occurring randomly between 3 to 5 sec)

Results

The results of this two-case study are summarized in table 2. Analyzing these results, one can state that the defender with the SAW has the following advantages over the defender with the M16 rifle:

1. The survivability probability increases from 42% to 50%.

2. The attackers are stopped at a farther range (418 vs 362 m) within a shorter time period (44 vs 75 sec).

Based on the very limited number of replications, the defender's ammunition expenditure is identical for the two cases when the defender survives.

In conclusion, the dynamic assault model is operable and capable of evaluating the effectiveness of individual small caliber weapons and weapon mixes in both the defensive and assault modes.

CONCLUSIONS

The BREACH simulation language exemplifies a modeling philosophy applicable to most types of weapon systems analysis studies requiring high-resolution combat scenarios. BREACH simulations offer the potential for analyzing the effectiveness of existing, developmental, or conceptual weapon systems in virtually any environment.

Table 2. Dynamic duel results

		Case 1	Case 2
		SAW	M16
P	(Red)	42	50
T	(Red)	37	63
T	(Blue)	44	75
R	(Blue)	418	362
T	(Blue 1)	30	58
R	(Blue 1)	457	411
T	(Blue 2)	60	101
R	(Blue 2)	374	287
NB	(Blue)	18	23
NB	(Red)	13	21
NB	(Red) when red survives	15	25

where

Red - defender

Blue - attacker

P - percentage of time killed

T - avg time of kill (sec)

R - avg range of kill (m)

NB - number of bursts

RECOMMENDATIONS

Recently, considerable interest in the BREACH simulation language has been shown throughout the systems analysis community, including DARCOM, ERADCOM, AMSAA, USAIS, and other ARRADCOM organizations. Lines of communication should be maintained and expanded throughout the community to most effectively employ this valuable analysis tool in the best interest of the US Army.

Consideration should be given within ARRADCOM to obtaining the latest version of the BREACH language and making it compatible with our existing computers.

REFERENCES

1. "A Battlefield Simulator of Weapon/Target Interaction (BREACH)," Vol I - Programmer Manual, Report Number 61, JTCG/ME-76-17-1, September 1976

2. "A Battlefield Simulator of Weapon/Target Interaction (BREACH)," Vol II - Users Manual, Report Number 61, JTCG/ME-76-17-2, September 1976

3. "A Battlefield Simulator of Weapon/Target Interaction (BREACH)," Vol III - Analyst Manual, Report Number 61, JTCG/ME-76-17-3, September 1976

4. P.D. Taska and J. D. Rouse, "BREACH," Vol 1, Programmer Manual - Version Eight, IITRI Project J6406, IIT Research Institute, Chicago, IL, February 1977

5. L. Barbarek, "BREACH User Manual," Version-8, IIT Research Institute, Chicago, IL, 1977

6. L. Barbarek, "Battlefield Related Electronic Warfare Simulation (BREWS) Advance Computer Programming Language Manual," Vol I - Users Manual, Technical Report ECOM-76-1955-U, US Army Electronics Command, Fort Monmouth, NJ, April 1977

7. P. Taska, T. Banaresh, J. Shields, and L. Barbarek, "Battlefield Related Electronic Warfare Simulation (BREWS) Advance Computer Programming Language Manual," Vol II - Programmer Manual, Technical Report ECOM-76-1955-P, US Army Electronics Command, Fort Monmouth, NJ, May 1977

8. L. Barbarek, G. Ebey, "Battlefield Related Electronic Warfare Simulation (BREWS) Advance Computer Programming Language Manual," Vol III - Analyst Manual, Technical Report ECOM-76-1955-A, US Army Electronics Command, Fort Monmouth, NJ, June 1977

9. H. K. Fallin, Jr., "Analysis of Machine-Gun Burst Dispersion Data with Corresponding Effectiveness Model," Technical Memorandum No. 33, US Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD, July 1969

10. A. D. Groves, "The Mathematical Analysis of a Simple Duel," Ballistic Research Laboratories Report No. 1261, August 1964

BIBLIOGRAPHY

"Advanced Firepower Concepts for Military Operations in Built-Up Areas," (2 volumes), ARPA Order Number 2163, Ketron, Inc., Arlington, VA, September 1973

Alexander, A. and Goss, C., "Evaluation Testing of SMAW Candidate Warheads Against Bunker and Urban Targets," Report Number NSWC/DL TR-3412, Naval Surface Weapons Center, Dahlgren, VA, October 1977

"Analysis of Munitions Effectiveness in Built-Up Areas Overseas," First Summary Report - Vol 1, Ketron, Inc., Arlington, VA, April 1973

Banash, R. "FAMAS A Small Unit Combat Simulation (Preliminary)," Report Number OR-67-3, Weapons Operations Research Office, US Army Weapons Command, Rock Island, IL, October 1967

Barbarek, L. "BREACH - Battlefield Related Evaluation and Analysis of Concepts and Hardware," Handout for DARCOM Briefing, IIT Research Institute, Chicago, IL, July 1977

"Civil Disturbances," Field Manual 19-15, HQ, Dept of the Army, Washington, DC, March 1972

"Combat in Cities Report," (3 volumes), US Army Infantry School, Fort Benning, GA, October 1972

Donnely, C. N., "Soviet Techniques for Combat in Built-Up Areas," <u>Defense Review</u>, Vol 10, April 1977

Eckhardt, W. and Mader, D. "CARMONETTE III Documentation," Volume 1, General Description, Report Number RAC-R-28, Research Analysis Corp., McLean, VA, October 1967

Ellefsen, R., Ph, D., Coffland, B. and Orr, G., "Urban Building Characteristics," Report Number NWSC/DL TR-3714, Naval Surface Weapons Center, Dahlgren, VA, January 1977

"Gaming Models for Military Operations in Built-Up Areas," (3 volumes), Report Numbers KFR-102/103/104-77, Ketron, Inc., Arlington, VA, November 1976

Goldberg, S. et al, "ASARS Battle Model," Book 1, Volume 1, Executive Summary, SA Group Technical Report TR 9073, Fort Belvoir, VA, May 1973

Manata, J. and Long, R., "Urban Warfare a First Report," Report Number R-TR-75-047, General Thomas J. Rodman Laboratory, Rock Island Arsenal, IL, October 1975

Marshall, S. L. A., "Notes on Urban Warfare," Special Publication No. 6, US Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, MD, April 1973

Mullen, W. and Shank, E., "The use of the 40 mm Grenande Launcher in a MOBA Environment," Technical Memorandum 20-78, US Army Human Engineering Laboratory, Aberdeen Proving Ground, MD, July 1978

"Symposium on Combat in Urban Areas," US Army Munitions Command, Picatinny Arsenal, Dover, NJ, March 1973

"Weapons Effects in Cities," (2 volumes), ARPA Order No. 2735, Intrec, Inc., Santa Monica, CA, December 1974

APPENDIX

BREACH CODE FOR STATIC DUELS

The BREACH programming code for static duel simulation is shown in figure A-1. This coding is included to illustrate the commands in a BREACH combat simulation model. This coding does not show the full capability of the BREACH language due to the simplicity of the model simulated, nor is this coding example presented as an illustration of perfect BREACH programming technique. It is included so that the reader may become acquainted with the structure and commands.

A description of the scenario of the model was included in the static duel section of the report. A flowchart showing the logic of the static dual is included in figure A-2. The main coding effort was directed at keeping track of firing sequences and the combatant's status. Time-of-flight was considered to allow for the possibility of duel kills.

The BREACH program language is based on a file, or program segment system, as described in the BREACH section of the report. The control deck for the static duel on the CDC 6600 is shown in figure A-3. The ATTACH commands are CDC SCOPE Commands which access the BREACH compiler, the referenced code for the static duel (Tape 14), and the input variables (Tape 11). See figure A-4. The INCLUD executive phase command executes the referenced file. Commands 170 to 200 include the referenced files into the computer run stream. The included files are the BREACH static duel file (14), input parameter (11), and file 28.

File 28 is written on file 14 between statement 980 (COPY, 28, R) and 1060 (END). The COPY-END statements set up a file with all the statements between the COPY and END included in that file. File 28 initializes the input parameters as well as the location and status of the combatants (via RESET, V). This file includes file 29 listed in lines 440 to 920 of the static duel code. File 29 contains the code that corresponds to the logic flowchart for the static duel.

A brief description of the static duel code is as follows: Statements 130 to 150 set up a one-cell map 100 meters square. The MOBILI, BREACH and VEGETA parameters are required by the BREACH environment phase; however, they are not used in the static duel model. Therefore, the statements listed 160 to 250 are included with dummy parameters. The combatants are described in statements 260 to 410. The parameter list following the MACHINE command corresponds to such dimensions as width, length, height, weight, etc. The VWIDTH and VULNERABILITY commands specify the vehicle vulnerability zones and action level in each vulnerability zone to damage the vehicle. Again these commands are required by BREACH (Vehicle Phase) but not used by the static model. The I Command, for inventory, sets the combatants name and total number in the program. The above statements are on file 14 and are executed only once for a program run.

The PATH command (line 450) on file 29 generates a path using the coordinates given and builds an event table which contains geographical information as well as firing event and mine encounter events. The FIRE command describes a direct or indirect firing table of times, distances, hit points, and probabilities of kill. In line 460 Blues fire table is set up. There are 10 firing events starting at time F10 uniformally distributed between F6 and F7 with a probability F2 against RED. The MOVE command moves a vehicle along a path as defined by a PATH command. In line 480 RED number 1 is moved on path with a velocity of 0.03 meters per second with the fire table BLUEKILL and REDKILL in effect.

The remaining commands in the static duel code are FORTRAN-like in syntax. For example, GOTO skips down to the referenced label before resuming execution. SETF sets the referenced flag with the listed value. SETF can also be used to perform mathematical operations. For example, in-line 810 MD (AN EXEC USER Cell) is set to 1. By using the user cells such as MD, ME, and MN, one can obtain moving averages via the SUBTOTALS command for multiple replications.

This static duel coding shows how easily BREACH coding may be changed for different input parameters and/or scenarios. Through the use of the file system in BREACH, sections of a simulation program can be modified independently.

STATIC DUEL - TAPE 14

```
100=NOLIST
110=OPERAT,T,I
120=SEED,.739563
130=LABEL,DMAP
140=ENVIRO
150=M,0.,0.,100.,100.,1,1
160=F,MOBILI,1,2
170=L,1,.0000000005,25.
180=SS,1,0.,0.,0.,100.,100.,100.,100.,0.,0.,0.
190=Z,1,1,1
200=F,BEACH,1,0
210=L,1
220=Z,1,1,1
230=F,VEGETA,1,0
240=L,1
250=Z,1,1,1
260=LABEL,PEOPLE
270=VEHICL
280=MACHINE,BLUE,MOBILI,1,1,.6,.3,1.8,.1,.25,.9,70.,.01,65.,.02
290=VWIDTH,.1,.1,.1,,,,0.
300=VULNER,0.,.1,.1,.1,.1,.1,.1,.1
310=I,BLUE,N=3
320=VEHICL
330=MACHINE,DBLUE,MOBILI,1,1,.6,.3,1.8,.1,.25,.9,70.,.01,65.,.02
340=VWIDTH,.1,.1,.1,,,,0.
350=VULNER,0.,.1,.1,.1,.1,.1,.1,.1
360=I,DBLUE,N=3
370=VEHICL
380=MACHINE,RED,MOBILI,1,1,.6,.3,1.8,.1,.25,.9,70.,.01,65.,.02
390=VWIDTH,.1,.1,.1,,,,0.
400=VULNER,0.,.1,.1,.1,.1,.1,.1,.1
410=I,RED,N=3
420=EXEC,V
430=COPY,29,R
```

Figure A-1. BREACH code for static duel.

```
440=EXEC,V
450=PATH,BLUE,XY=50.,50.,52.,50.
460=FIRE,D,REDKILL,FT=F10,U,F6,F7,NT=10,P=0.,0.,F2
470=FIRE,D,BLUEKILL,FT=F11,U,F8,F9,NT=10,P=F1,0.,0.
480=MOVE,RED,1,.03,FIRE=BLUEKILL,REDKILL $ BLUE FIRES
490=SETF,IF30,ST    $F30 HAS RED STATUS
500=SETF,F31,ET     $F31 HAS RED ENDTIME
510=MOVE,BLUE,1,.03  $ RED FIRES
520=$  ST IS BLUE STATUS
530=$  ET IS BLUE ENDTIME
540=OPERAT,T,B
550=GOTO,CONT,IF,V,BLUE,1,M,AND,V,RED,1,M  $ CONT IF BOTH ALIVE
560=GOTO,BRTEST,IF,V,BLUE,1,K,AND,V,RED,1,K$ BRTEST IF BOTH DEAD
570=GOTO,BDEAD,IF,V,BLUE,1,K
580=GOTO,RDEAD
590=LABEL,CONT
600=OPERAT,T,I
610=SETF,F10,F31,U,F6,F7  $BLUES NEW FIRETIME
620=SETF,F11,ET,U,F8,F9  $REDS NEW FIRETIME
630=OPERAT,T,B
640=GOTO,FINI
650=LABEL,BRTEST
660=GOTO,SKIP,IF,T,ET,GE31  $BTIME GE RTIME
670=$BLUE KILLED FIRST
680=SETF,F41,ET,F4          $F41=ET(B)+TOF(R)
690=SETF,XD,F31             $XD=DUEL ENDTIME
700=SETF,DT,F31
710=GOTO,BRDEAD,IF,T,XD,LE41
720=GOTO,BDEAD
730=LABEL,SKIP                   $RED KILLED FIRST
740=SETF,F40,F31,F3              $F40=ET(R)+TOF(B)
750=SETF,DT,,ET
760=GOTO,BRDEAD,IF,T,ET,LE40
770=GOTO,RDEAD
780=LABEL,RDEAD
790=OPERAT,T,I
```

Figure A-1. Continued.

```
00=SETF,DT,,F31                    $DUEL ENDTIME=RED ET
810=SETF,MD,,1.                     $MD IS RED COUNTER
820=EFFECT
830=RETURN
840=LABEL,BDEAD
850=OPERAT,T,I
860=SETF,DT,ETT                     $DT IS BLUE ENDTIME
870=SETF,ME,,1.                     $ME IS BLUE COUNTER
880=EFFECT
890=RETURN
900=LABEL,BRDEAD
910=OPERAT,T,I
920=SETF,MN,,1.                     $MN IS BLUE+RED COUNTER
930=EFFECT
940=RETURN
950=LABEL,FINI
960=OPERAT,T,I
970=END
980=COPY,28,R
990=EXEC,V
1000=SETF,F10,,0.
1010=SETF,F11,,F5
1020=RESET,V
1030=INCLUD,29,R,9
1040=SUBTOTAL
1050=EXEC,V
1060=END
..
```

Figure A-1. Continued.

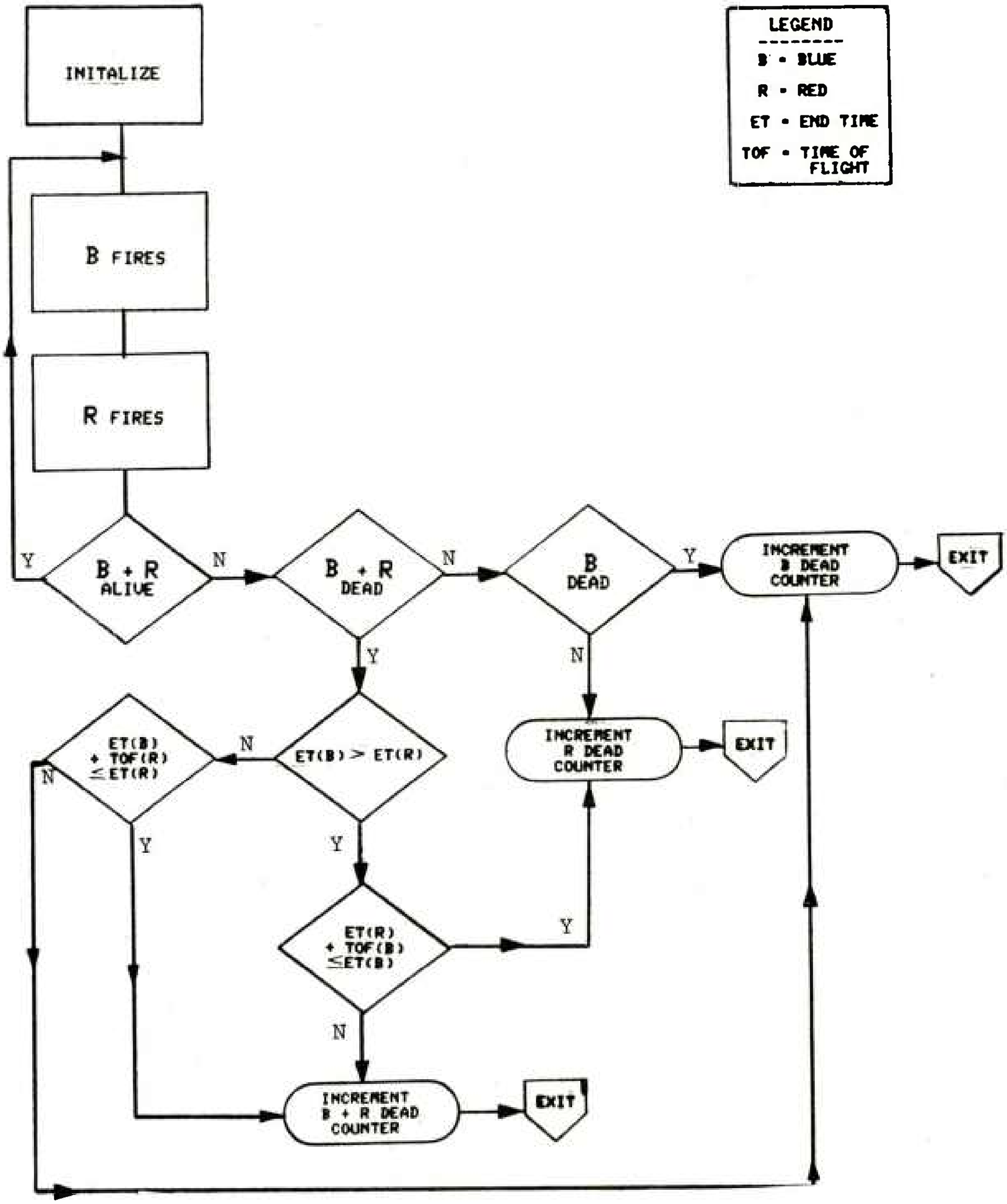

Figure A-2. Flowchart and logic for static duel.

```
100=LNBS,CM130000,T500,IO500
110=COMMENT.(XXX-XXX,12345A)SCORS
120=ATTACH,XX,BRE7SHORT,ID=DRDARSCSM.
130=ATTACH,TAPE14,ID=SCORS.
140=ATTACH,TAPE11,ID=SCORS.
150=XX.
160=*EOR
170=INCLUD,14,R
180=INCLUD,14
190=INCLUD,11,R
200=INCLUD,28,R,100
210=*EOR
220=*EOF
```

Figure A-3. Program control card images for static duel.

TAPE 11

```
100=SETF,F1,10.         $PROBABILITY OF BLUE HIT
110=SETF,F2,24.         $PROBABILITY OF RED HIT
120=SETF,F3,1.          $TIME OF FLIGHT OF BLUE'S ROUND
130=SETF,F4,1.          $TIME OF FLIGHT OF RED'S ROUND
140=SETF,F5,10.         $RED'S REACTION TIME
150=SETF,F6,1.          $F6 AND F7 ARE PARAMETERS DESCRIBING
160=SETF,F7,2.          $ BLUE'S RATE OF FIRE
170=SETF,F8,1.          $RED'S RATE OF FIRE PARAMETERS
180=SETF,F9,2.
190=SETF,F10,10.        $RED'S INITIAL FIRE TIME
200=SETF,F11,20.        $BLUE'S  INITIAL FIRE TIME
```

Figure A-4. Input parameters for static duel.

DISTRIBUTION LIST

Commander
US Army Armament Research and Development Command
ATTN: DRDAR-SC, COL Larkins
DRDAR-SC Dr. Gyorog
DRDAR-SCS, LTC Bourgeois
DRDAR-SCS-M, Mr. Cohen
Mr. Kwatnoski (10)
DRDAR-SCS-E, Mr. Ackley
Mr. Mancini
DRDAR-SCA-T, Mr. Kahn
Mr. Steiner
DRDAR-SCA-W, Mr. Gehbauer
DRDAR-SCA-C, Mr. Gadomski
DRDAR-SCA-A, Mr. McHugh
Mr. Schlenner
DRDAR-SCA-S, Mr. Rhoades
DRDAR-SCF-D, Mr. Goodman
DRDAR-SCF-A, Mr. Staton
DRDAR-SCF-C, Mr. Langan
DRDAR-SA, Mr. Stearns
DRDAR-AC, LTC Hackley
Mr. Fairbanks
DRDAR-SEA, Mr. Weir
Mr. Pearcy (3)
DRDAR-MS, Mr. Grobstein
DRDAR-MSA, Mr. Isakower
Mr. Tobak (3)
DRDAR-MSM, Mr. Barnett
DRDAR-MST, Mr. Rucker
DRDAR-LCW, Mr. Garver
DRDAR-LCS, Mr. Gregorits
Dr. Einbinder
DRDAR-LCU, Mr. Strano
Mr. Costa
DRDAR-TSS (5)
DRCPM-SA, Mr. Smith
DRCPM-NUC
DRCPM-CAWS
DRCPM-TMA
DRCPM-ADG
Dover, NJ 07801

Office, Deputy Under Secretary of the Army (OR)
The Pentagon
Washington, DC 20310

Commander
US Army Materiel Development
and Readiness Command
ATTN: DRCBSI-I, Dr. G. Anderson (2)
DRCPA-S, Mr. D. Sen (2)
DRCGS-F, MAJ M. Janay (2)
5001 Eisenhower Avenue
Alexandria, VA 22333

Commander
US Army Tank-Automotive R&D Command
Warren, MI 48090

Commander
US Army Aviation Systems Command
St. Louis, MO 63166

Commander
US Army Electronics R&D Command
ATTN: DRDEL-ST-T, Dr. B. Zarwyn
DRDEL-OA
2800 Powder Mill Road
Adelphi, MD 20783

Commander
US Army Missile R&D Command
ATTN: Advanced Systems Concepts Office
Redstone Arsenal, AL 35809

Commander
US Army Mobility Equipment R&D Command
ATTN: DRDME-VT, Mr. G. Bryan
Fort Belvoir, VA 22060

Commandant
US Army Military Academy
West Point, NY 10996

Commandant
US Army Air Defense School
Fort Bliss, TX 79906

Director
US Army Human Engineering Laboratory
ATTN: DRXHE-SPG, Mr. E. Shank
DRXHE-SPG, Mrs. B. Thein
Aberdeen, MD 21005

Commander (12)
US Army Defense Documentation Center
Cameron Station
Alexandria, VA 22314

Commander
CORADCOM
ATTN: DRDCO-PPA-SA, Mr. Irwin
Fort Monmouth, NJ 07703

Commander
US Army Infantry School
ATTN: ATSH-CD-CS, Ms. Shirley (2)
ATSH-ID, CPT Carlson
Fort Benning, GA 31905

Commander
US Army Training & Doctrine Command
Fort Monroe, VA 23351

Project Management, Training Devices
Naval Training Equipment Center
Orlando, FL 32813

Commander
TRADOC Systems Analysis Activity
ATTN: ATAA-TEM (2)
White Sands Missile Range, NM 88002

Commander
US Army Combined Arms Combat
Development Activity
ATTN: ATCA-CA, R. Dekinder (2)
Fort Leavenworth, KS 66027

Commander
US Army Armament Materiel Readiness Command
ATTN: DRSAR-SA (2)
Rock Island, IL 61201

Commander
US Army Research Office
PO Box 12211
Research Triangle Park, NC 27709

Commandant
US Army Field Artillery School
Fort Sill, OK 73503

Commandant
US Naval War College
Center for War Gaming
Newport, RI 02840

Commandant
Naval Postgraduate School
Dept of Operations Analysis
Monterey, CA 93940

Commander
US Marine Corps Development
& Education Command
ATTN: Plans & Studies Division
Quantico, VA 22314

Office of Naval Research
Code 455
Washington, DC 20360

Naval Weapons Center
ATTN: Technical Library, Code 753
China Lake, CA 93555

Naval Coastal Systems Center
ATTN: Code 772
Panama City, FL 32407

Commander
Air Force Armament Laboratories
ATTN: DLYW, Mr. C. Reynolds
Eglin Air Force Base 32542

Institute for Defense Analysis
400 Army-Navy Drive
Arlington, VA 22202

Battelle Columbus Laboratories
505 King Avenue
Columbus, Ohio 43201

Analytics
ATTN: Mr. George Schecter
Suite 540
7926 Jones Branch Drive
McLean, VA 22101

IIT Research Institute
ATTN: Mr. Barbarek (3)
10 West 35th Street
Chicago, IL 60616

Commander
USAACFT
ATTN: ATZQ-D-CS, CPT Olson
Fort Rucker, Dothan, AL 36362

Commander
9th Division
ATTN: AFZH-DG3
Fort Lewis, Tacoma, WA 98433

Commander
Berlin Brigade G3
ATTN: LTC R. C. Becker
APO
New York 09742

Commander
Defense Advanced Research Projects Agency (2)
1400 Wilson Road
Arlington, VA 22209

Commander
US Army Concepts Analysis Agency
8120 Woodmont Avenue
Bethesda, MD 20014

Commander/Director
Chemical Systems Laboratory
ATTN: DRDAR-CLY, Mr. S. Meyer
Aberdeen Proving Ground, MD 21010

Commander
US Army Combat Developments
Experimentation Command
ATTN: ATEC-PL-PA
Fort Ord, CA 93941

Director
Electronic Warfare Laboratory
ATTN: DELEW-P, Dr. DeVilbiss
Ft. Monmouth, NJ 07703

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GI, Mr. Clifford
Mr. Walker
DRXSY-GA
DRXSY-GP
DRXSY-GB
DRXSY-C
DRXSY-M
Aberdeen Proving Ground, MD 21005

Director
Ballistics Research Laboratory
ATTN: DRDAR-BLB, Mr. Johnson
Mr. Hirschberg
DRDAR-BLV
Aberdeen Proving Ground, MD 21005

Commander
US Army Armament Materiel and
Readiness Command
ATTN: DRSAR-LEP-L
Rock Island, IL 61299

Director
US Army TRADOC Systems
Analysis Activity
ATTN: ATAA-SL (Tech Lib)
White Sands Missile Range, NM 88002

U.S. Army Materiel Systems Analysis Activity
ATTN: DRXSY-MP
Aberdeen Proving Ground, MD 21005

Weapon System Concept Team/CSL
ATTN: DRDAR-ACW
Aberdeen Proving Ground, MD 21010

Technical Library
ATTN: DRDAR-CLJ-L
Aberdeen Proving Ground, MD 21005

Technical Library
ATTN: DRDAR-TSB-S
Aberdeen Proving Ground, MD 21010

Technical Library
ATTN: DRDAR-LCB-TL
Benet Weapons Laboratory
Watervliet, NY 12189